

贪心算法

与算法导论(第三版)章节的对应关系

■ Chapter 16. 贪心算法

■ 涉及知识点包括:

- ❖ 引子: 找零问题
- ❖ 贪心算法的特性
- ❖ 贪心算法的应用



贪婪，我找不到一个更好的词来描述它，
它就是好！它就是对！它就是有效！

——美国演员迈克尔·道格拉斯，影片《华尔街》

引子：找零问题 (Change Making Problem)

- 问题描述：给定无限数量的面额硬币 m 种， $d_1 > d_2 > \dots > d_m$

请用**最少数量**的硬币找出金额为 n 的零钱。

- 例如硬币面额为： $d_1 = 25, d_2 = 10, d_3 = 5, d_4 = 1$ ，(单位：cent)

人们一般都选择采取贪心算法找零，即每一步都选出满足要求的最大面额的货币。

若 $n=48$ cents，下意识如何做：

1个 25 cents的硬币

2个 10 cents的硬币

3个 1 cent的硬币

思考：这个解是否是最优的？

- 可以证明，对于上例给定的硬币面额，所有任意正整数金额，贪心算法都可以给出一个最优解

- 思考：任意给出一组钞票面值，判断是否能够通过贪心算法得到最优解

反例：钞票面额为 7, 5, 1, $n=10$,

贪心算法解策略： $10 = 7+1+1+1$

最优解策略： $10 = 5+5$

- 结论：并不是所有问题都可以通过贪心算法得到最优解

- 贪心算法的核心思想：贪心算法通过一系列步骤构造问题的解，每一步对当前构造的部分做一个扩展，直到获得完整的解

贪心算法

通过一系列选择逐个构建优化问题的解决方案，这些选择是：

- Feasible（可行的）
- Locally optimal（局部最优）
- Irrevocable（不可撤销的）

对于某些问题，为每个实例产生一个最佳解决方案，对于部分问题，虽不能产生最优解，但却能够快速逼近最优解

贪心算法

■ 一般贪心算法具有如下特性：

- ❖ 优化问题，有一个候选对象的集合，如硬币，边(Kruskal)，路径(Dijkstra)，顶点(Prim)等；
- ❖ 随着算法的进行，累积行成两个集合，一个是已经被选中的对象集合，另一个是被抛弃的对象集合；
- ❖ 有一个函数(solution function)，该函数用于检查候选对象集合是否是（or 提供了）问题的解，该函数不考虑此时的解决方法是否是最优的；

- ❖ 还有一个函数，检查候选对象(candidate)是否可加入到当前解的对象集合中(*i.e.*, feasible可行的)，同样，这一步不考虑解决方案的最优性；
- ❖ **选择函数(selection function)**，指出哪个剩余的候选对象（没有被选择过也没有被丢弃过）最有可能构成问题的解；
- ❖ **目标函数(object function)**，给出解的值。如硬币个数，路径长度，顶点个数等。

贪心算法

- 求最优解的问题可看作是通过一系列步骤，每一步有一个选择的集合，对于较简单的问题，动态规划显得过于复杂，可用较简单有效的算法求解之。
- 贪心算法总是在当前步骤上选取最好的方案，即它是一种**局部最优**的选择，并希望它导致一个全局最优，但有时是(或者是大部分)不可能导致**全局最优**。
- 例：求 v_i 到 v_j 的一条最短路径，若从 v_i 搜索到邻点 v_k 最短，未必是 v_i 到 v_j 最短。
- 但是，仍有许多问题贪心法将产生全局最优解，如**MST**，**单源最短路径**等。

贪心算法的应用

■ 调度问题

- ❖ 活动选择 (Chap 16.1)
- ❖ Minimizing time in system
- ❖ 用拟阵问题求解任务调度(Chap 16.5)

■ 图

- ❖ 最小生成树(Chap 23)
- ❖ Dijkstra最短路径算法 (Chap 24.3)

■ 其他问题

- ❖ 哈夫曼编码 (Chap 16.3)
- ❖ 图片着色问题
- ❖ 旅行商问题 (Chap 35.2)
- ❖ 集合覆盖问题 (Chap 35.3)
- ❖ 子集和问题 (Chap 35.5)